

Как "добиться" времени записи 200 ms на NVMe-дисках
Андрей Копейко
andrey+hl2022@kopeyko.ru
HL
HighLoad++
2022

В начале был вопрос...

У нас Postgres стал "тормозить", можете полечить?



Что говорит мониторинг?



Что говорит мониторинг?



А как должно быть?

Тип диска	Типичное время записи, ms
HDD SATA 5400 rpm	7 - 15
HDD SATA 7200 rpm	5 - 10
SSD SATA TLC	1 - 2
NVMe U.2 TLC	0.05 - 0.1



8 шагов "к успеху"



8 шагов "к успеху"

- Каждый шаг в отдельности — абсолютно правилен и логичен, и узаконен рекомендациями соответствующего вендора.

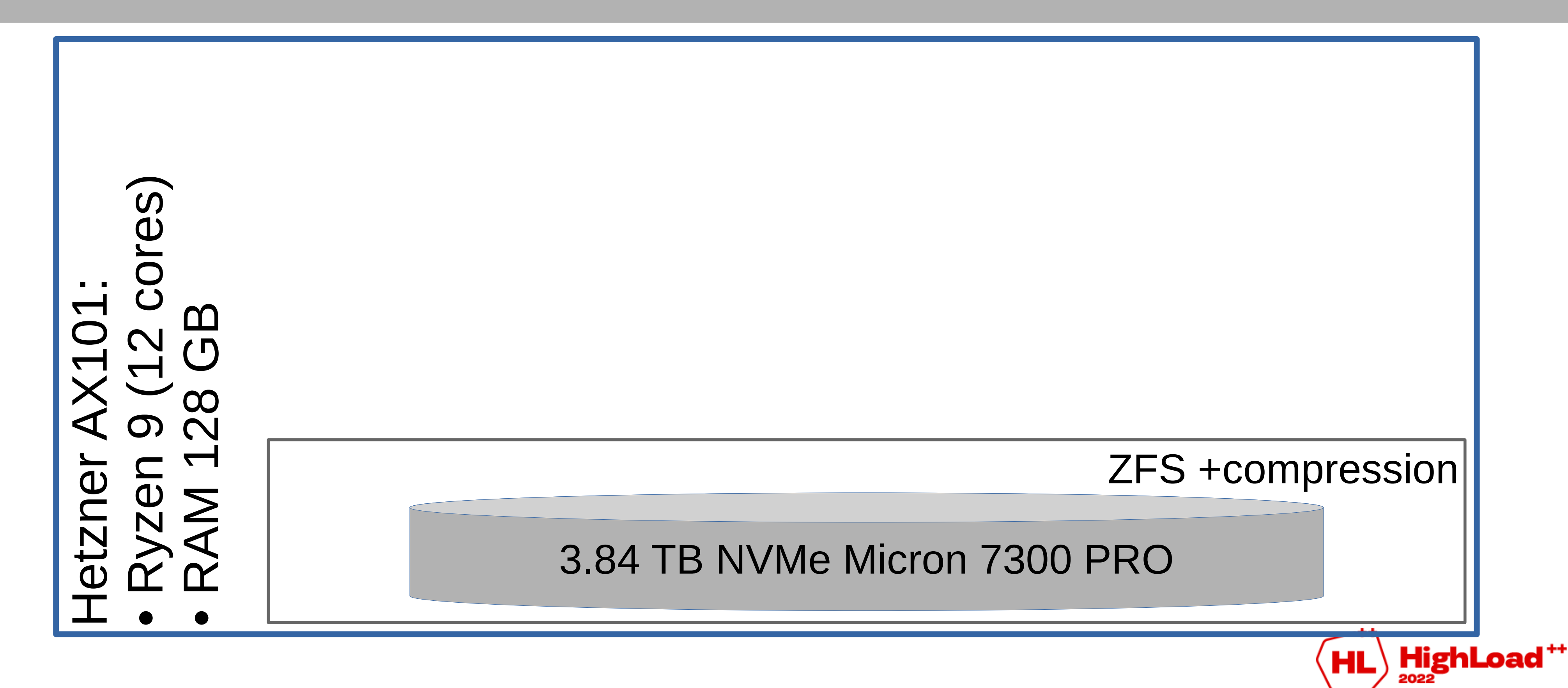


8 шагов "к успеху"

- Каждый шаг в отдельности — абсолютно правилен и логичен, и узаконен рекомендациями соответствующего вендора.
- Итоговый результат - получается «так себе...»



1/8 шагов "к успеху"



2/8 шагов "к успеху"

Hetzner AX101:
- Ryzen 9 (12 cores)
- RAM 128 GB

Proxmox

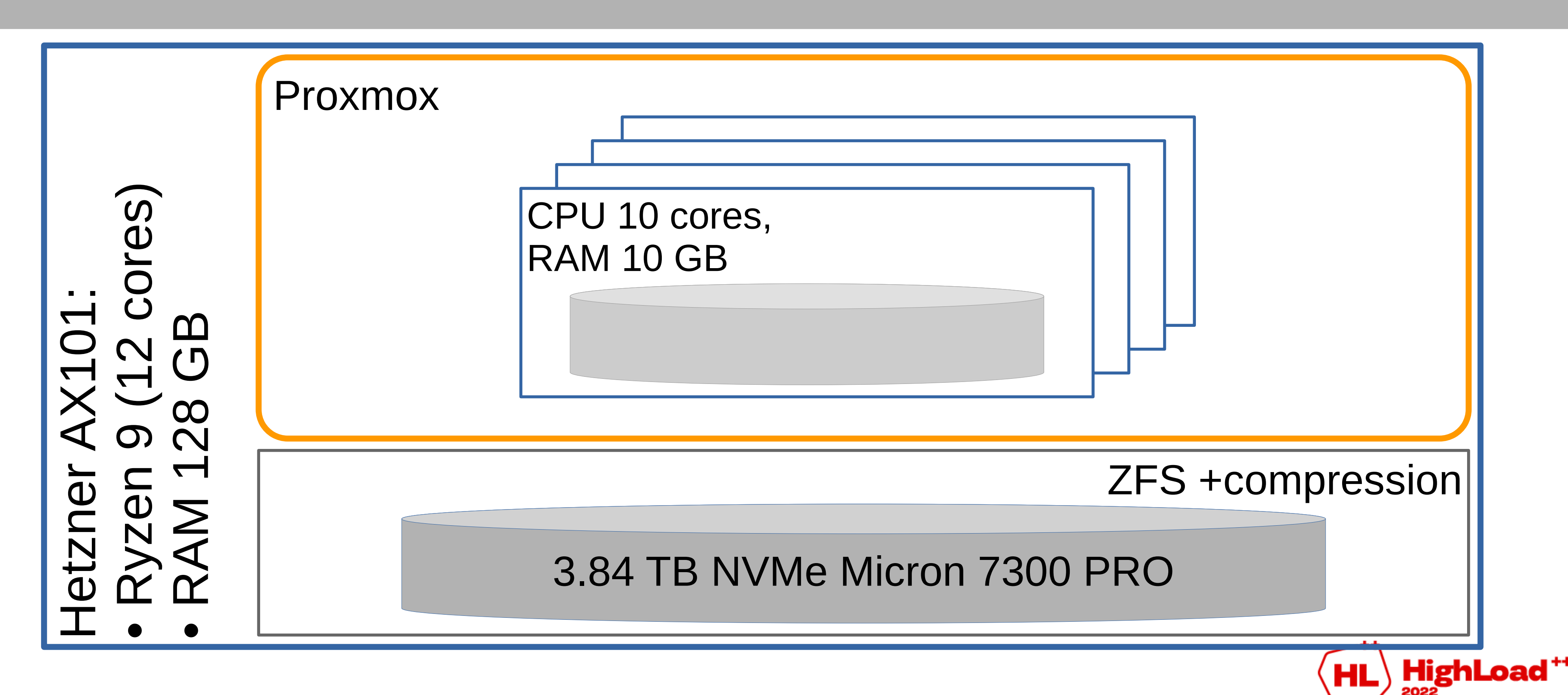
- *диски для всех KVM-виртуалок создаём на ZFS пуле*

ZFS +compression

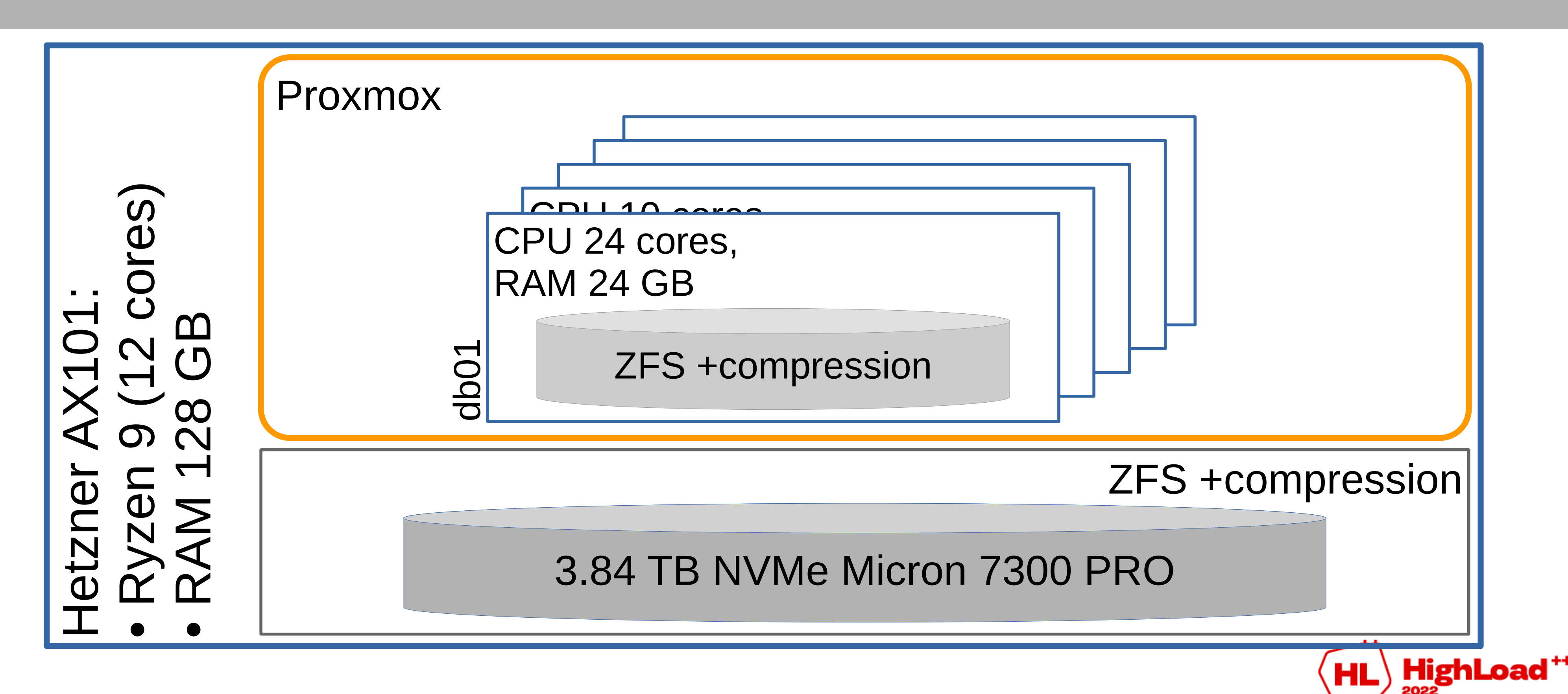
3.84 TB NVMe Micron 7300 PRO



3/8 шагов "к успеху"



4/8 шагов "к успеху"



5/8 шагов "к успеху"

CPU 24 cores,
RAM 24 GB

+

ZFS +compression

db01



6/8 шагов "к успеху"

```
db01=# CREATE TABLE table1 (
  time TIMESTAMPTZ NOT NULL,
  device_id INTEGER NOT NULL,
  cpu DOUBLE PRECISION);
db01=# SELECT
create_hypertable('table1','time');
```



Hypertables & chunks

Normal table

time	value
2021-01-02 00:00:00	36
2021-01-02 06:00:00	5
2021-01-02 23:00:00	29
2021-01-03 00:00:00	17
2021-01-03 06:00:00	8
2021-01-03 23:00:00	6
2021-01-04 00:00:00	41
2021-01-04 06:00:00	14
2021-01-04 23:00:00	5

Hypertable

time	value
Chunk ID 1	
2021-01-02 00:00:00	36
2021-01-02 06:00:00	5
2021-01-02 23:00:00	29
Chunk ID 2	
2021-01-03 00:00:00	17
2021-01-03 06:00:00	8
2021-01-03 23:00:00	6
Chunk ID 3	
2021-01-04 00:00:00	41
2021-01-04 06:00:00	14
2021-01-04 23:00:00	5



7/8 шагов "к успеху"

```
db01=# ALTER TABLE table1 SET (
  timescaledb.compress,
  timescaledb.compress_orderby =
'time DESC');
db01=# SELECT
add_compression_policy('table1',
INTERVAL '3 days');
```

Сжатие в Timescale

Документация
https://docs.timescale.com/timescaledb/latest/overview/core-concepts/compression/
прельщает и завлекает:

«In tests, TimescaleDB achieves 91-96% storage savings with lossless compression. This equals a compression ratio between 10 and 23. For comparison, a compressed file system, such as ZFS or BTRFS, usually achieves 3 to 9 times compression.»



Сжатие в Timescale

time	device_id	cpu
12:00:02	1	88.2
12:00:01	1	88.6
12:00:02	2	300.5



Сжатие в Timescale

time	device_id	cpu
12:00:02	1	88.2
12:00:01	1	88.6
12:00:02	2	300.5

time	device_id	cpu
[12:00:02, 12:00:02, 12:00:01]	[1, 2, 1]	[88.2, 300.5, 88.6]



Сжатие в Timescale

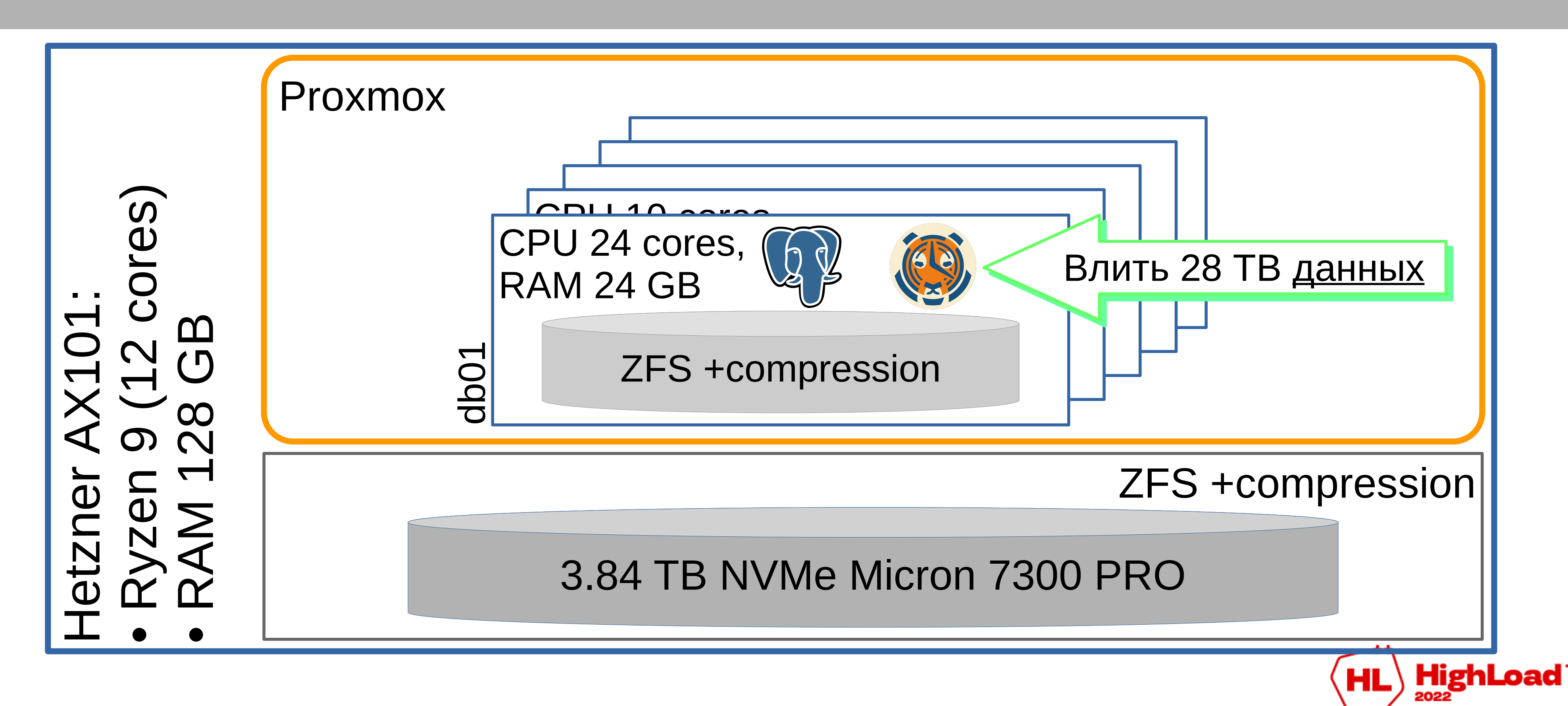
time	device_id	cpu
12:00:02	1	88.2
12:00:01	1	88.6
12:00:02	2	300.5

«TimescaleDB can combine up to 1000 entries into a single row.»

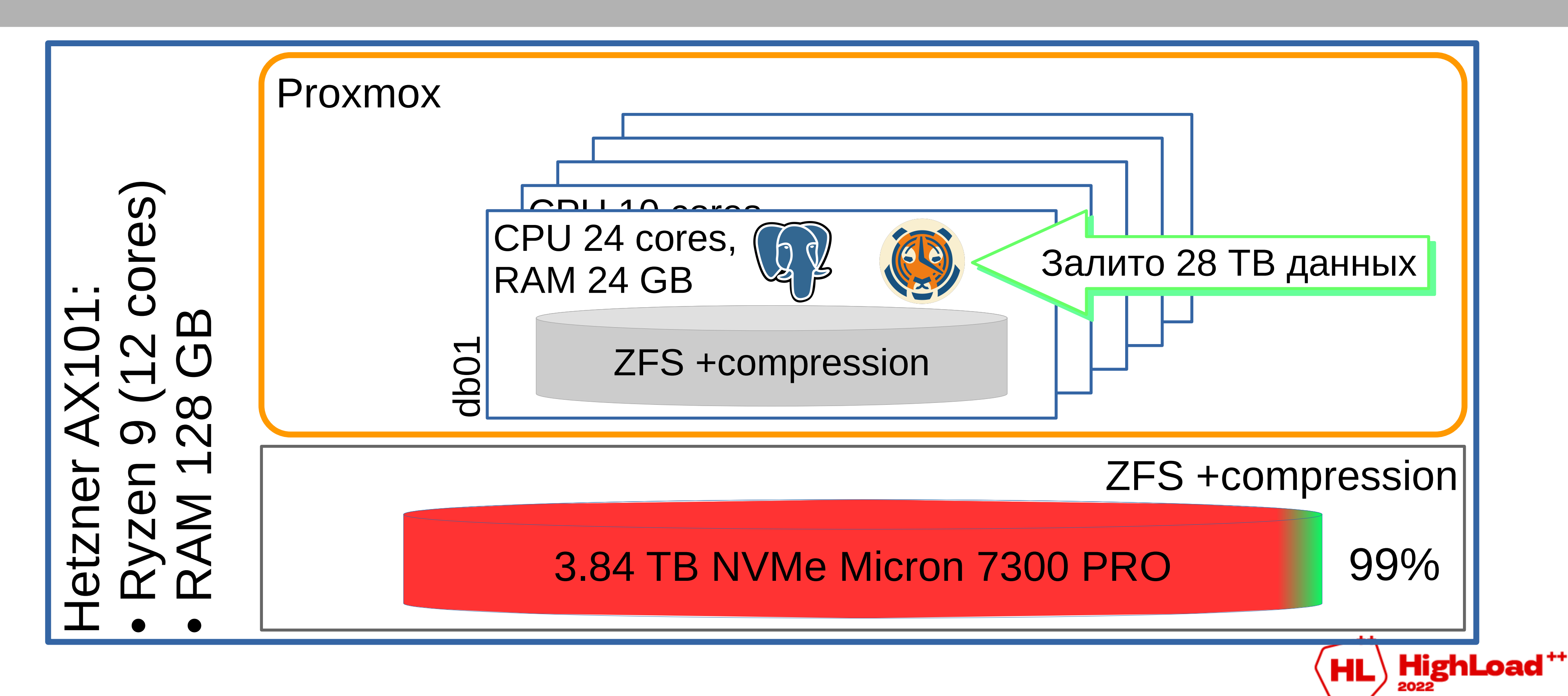
time	device_id	cpu
[12:00:02, 12:00:02, 12:00:01]	[1, 2, 1]	[88.2, 300.5, 88.6]



8/8 шагов "к успеху"



8/8 шагов "к успеху"

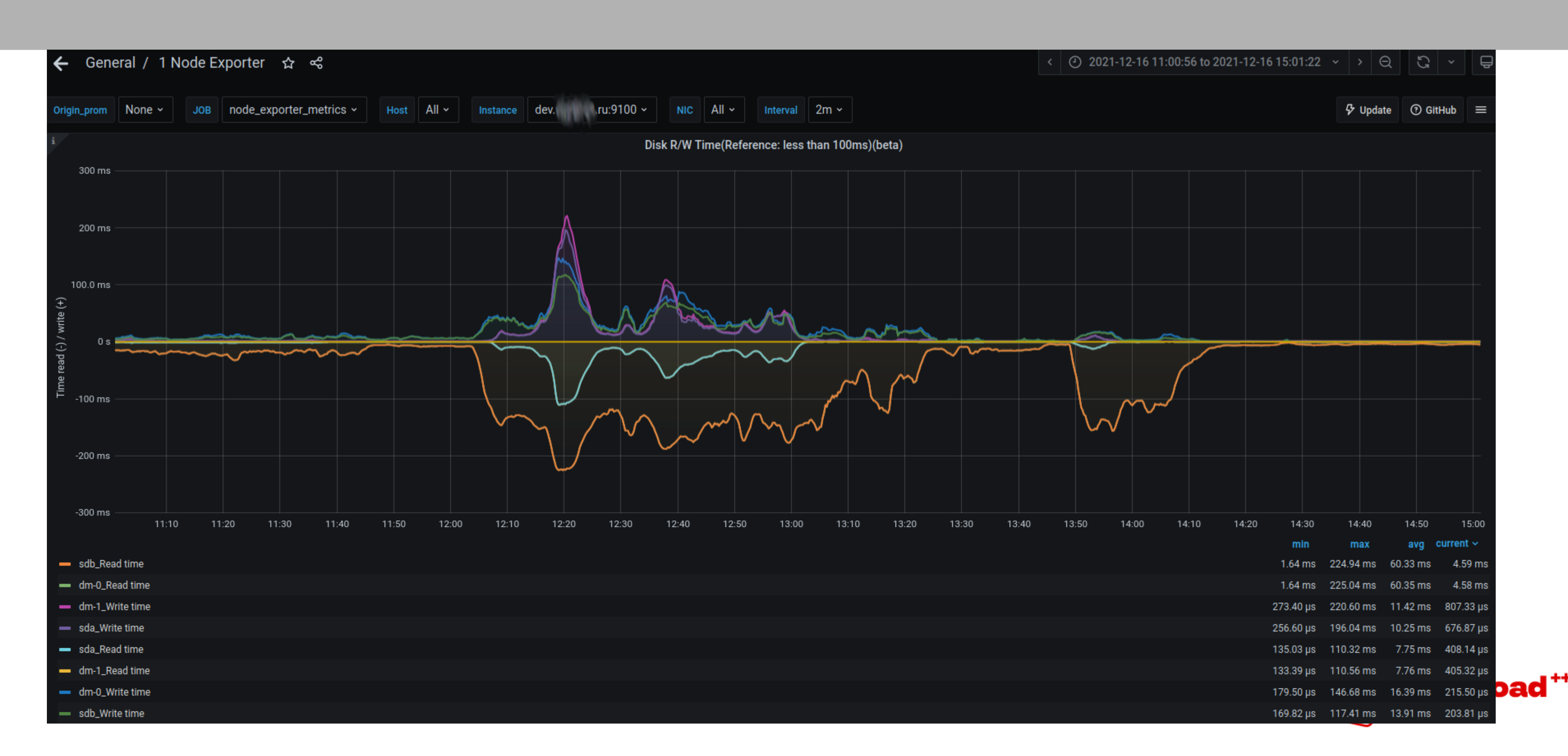


"Успех"!

```
SELECT * FROM table1
WHERE time > now() - INTERVAL '1
week';
```



"Успех"!


General / 1 Node Exporter
2021-12-16 11:00:56 to 2021-12-16 15:01:22
Origin_prom None
JOB node_exporter_metrics
Host All
Instance dev. .ru:9100
NIC All
Interval 2m
Update
GitHub
Disk R/W Time(Reference: less than 100ms)(beta)
300 ms
200 ms
100.0 ms
0 s
-100 ms
-200 ms
-300 ms
Time read (-) / write (+)
11:10 11:20 11:30 11:40 11:50 12:00 12:10 12:20 12:30 12:40 12:50 13:00 13:10 13:20 13:30 13:40 13:50 14:00 14:10 14:20 14:30 14:40 14:50 15:00
min max avg current
sdb_Read time 1.64 ms 224.94 ms 60.33 ms 4.59 ms
dm-0_Read time 1.64 ms 225.04 ms 60.35 ms 4.58 ms
dm-1_Write time 273.40 µs 220.60 ms 11.42 ms 807.33 µs
sda_Write time 256.60 µs 196.04 ms 10.25 ms 676.87 µs
sda_Read time 135.03 µs 110.32 ms 7.75 ms 408.14 µs
dm-1_Read time 133.39 µs 110.56 ms 7.76 ms 405.32 µs
dm-0_Write time 179.50 µs 146.68 ms 16.39 ms 215.50 µs
sdb_Write time 169.82 µs 117.41 ms 13.91 ms 203.81 µs

Что происходит?

Сжатие TOAST в PGSQL



Что происходит?

Сжатие TOAST в PGSQL

Сжатие ZFS в виртуалке



Что происходит?

Сжатие TOAST в PGSQL

Сжатие ZFS в виртуалке

Сжатие ZFS на хосте



Что происходит?

Сжатие TOAST в PGSQL

Сжатие ZFS в виртуалке

Сжатие ZFS на хосте

3.84 TB NVMe



Что происходит?

Сжатие TOAST в PGSQL

Сжатие ZFS в виртуалке

Сжатие ZFS на хосте

3.84 TB NVMe

200 ms



Что происходит?

```
andrey@server03:~# sudo smartctl --all /dev/nvme0n1 \
> |grep -e ^Model -e ^Total -e ^Data -e ^Power\ On
Model Number:                           MTFDHBE3T8TDF
Total NVM Capacity:                     3,840,755,982,336 [3.84 TB]
Data Units Read:                        3,116,110,545 [1.59 PB]
Data Units Written:                     7,523,806,027 [3.85 PB]
Power On Hours:                         7,379
andrey@server03:~#
```



Что происходит?

```
andrey@server03:~# sudo smartctl --all /dev/nvme0n1 \
> |grep -e ^Model -e ^Total -e ^Data -e ^Power\ On
Model Number:                       MTFDHBE3T8TDF
Total NVM Capacity:                 3,840,755,982,336 [3.84 TB]
Data Units Read:                    3,116,110,545 [1.59 PB]
Data Units Written:                 7,523,806,027 [3.85 PB]
Power On Hours:                     7,379
andrey@server03:~#
```

TBW = 9800

Warranty = 5 years



Выводы

- PostgreSQL не виноват



Выводы

- PostgreSQL не виноват
- Понимайте что происходит "под капотом"



Выводы

- PostgreSQL не виноват
- Понимайте что происходит "под капотом"
- Не заполняйте ZFS более 90%



Выводы

- PostgreSQL не виноват
- Понимайте что происходит "под капотом"
- Не заполняйте ZFS более 90%
- Конструкция неоперабельна...



У вас тоже тормозит PostgreSQL? Обращайтесь, поможем!

andrey+hl2022@kopeyko.ru

